

# TEMA 5. Sistemas Secuenciales Síncronos

1. Introducción. (2-5)
    - 1.1 Definición de sistema secuencial. Concepto de estado (6-10)
    - 1.2 Esquema general de un sistema secuencial. (11)
    - 1.3 Máquina de estados finitos. Autómata. (12-13)
  2. Autómatas de *Mealy* y *Moore*. (14-15)
    - 2.1 Modelo de *Mealy*. (16)
    - 2.2 Modelo de *Moore*. (17)
  3. Especificación y etapas en el diseño de sistemas secuenciales. (18-20)
    - 3.1 Tablas y diagramas de transición entre estados. (21-24)
    - 3.2 Ejemplos de sistemas secuenciales (25-39 )
  4. Implementación Canónica de sistemas secuenciales (40-41)
    - 4.1 Ejemplos de implementación (42-43)
  5. Especificación de SS mediante Diagramas ASM. (44- 45)
    - 5.1 Ejemplos de implementación (46-53)
  6. Modelado VHDL de máquinas de estados finitos (FSM) (46-61)
    - 6.1 Ejemplos de modelado VHDL (62-67)
  7. Análisis y Síntesis de Sistemas Secuenciales (68-70)
- 



## 1. INTRODUCCIÓN (I)

Dentro de los sistemas secuenciales, los **SISTEMAS SECUENCIALES SÍNCRONOS** es el caso más habitual y más sencillo, en el diseño práctico de dichos sistemas digitales

- **Sistemas síncronos:** aquellos en los que existe al menos una señal de reloj que va marcando en que momento se pueden producir los cambios en las salidas del circuito.
- Disparo de **reloj por flanco**.



## 1. INTRODUCCIÓN (II)

**EJEMPLO:** Circuito que no es combinacional: generador de bit de paridad

Constrúyase un circuito que nos indique, si el número total de 1's presentados en su entrada, hasta un instante determinado, es par o impar.



El sistema se puede especificar utilizando un tabla de verdad, en la que se reflejen las variables que intervienen en el sistema: **entrada, x; salida, z;** "situación del sistema en un instante"



3

## 1. INTRODUCCIÓN (III)

**EJEMPLO:** Circuito que no es combinacional:  
Generador de bit de paridad

TABLA DE VERDAD

	ENTRADA	SALIDA
Situación hasta el instante "t"	X(t)	Z(t)
Nº par de 1's	0 1	par impar
Nº impar de 1's	0 1	impar par



4

# 1. INTRODUCCIÓN (IV)

Será un sistema que tendrá:

1. Una única entrada para la llegada de datos en serie (X)
2. Una única salida que será el bit de paridad par, de la secuencia de bits llegados al sistema hasta el momento t (Z)
3. Dos estados internos, puesto que toda la historia pasada puede reducirse a dos situaciones:

- ❑ PAR (P) si el número de "1" llegados hasta el instante t es PAR
  - ❑ IMPAR (I) si el número de "1" llegados hasta el instante t es IMPAR
- Será necesaria 1 variable de estado y una función f que relacione el estado presente, con el estado anterior y la entrada presente de acuerdo con el esquema:

	X = 0	X = 1
PAR	PAR	IMPAR
IMPAR	IMPAR	PAR



## 1.1 DEFINICIÓN DE UN SISTEMA SECUENCIAL (I)

Un sistema es secuencial si la salida en cada instante t, depende tanto del valor de las entradas presentes en ese instante, como de la "historia" del sistema, estado esta determinada, por el estado del sistema en el instante que comienza a operar y por los valores de las entradas en ese momento inicial.

Un sistema secuencial depende de su "historia", pues de acuerdo de ella, el sistema estará en cada instante en un estado determinado



## 1.1 DEFINICIÓN DE UN SISTEMA SECUENCIAL (II)

1. **Dependen del tiempo:** para una misma combinación de entradas pueden obtenerse salidas diferentes; de esto, se deduce, que estas dependen del momento en que se presentan las distintas combinaciones de entrada.
2. **Debe existir realimentación de la salida a la entrada:** para que el estado del circuito repercuta en el valor de la salida, dicho estado ha de almacenarse y presentarse al circuito, como un "tipo de variable de entrada".



## 1.1 DEFINICIÓN DE UN SISTEMA SECUENCIAL (III)

En los sistemas secuenciales síncronos con disparo por flanco del reloj, está garantizado que:

1. Transcurrido un tiempo suficiente desde el flanco del reloj, los valores actualizados (**nuevos valores**) de las señales de entrada a la memoria del sistema (**biestables**) se propagarán a lo largo del mismo y éste adquirirá una configuración estable.

Las entradas al sistema deben ser estables y no deben existir lazos de realimentación puramente combinacionales (es decir sin que atraviesen una célula de almacenamiento, un biestable).

2. No sucederán cambios en el sistema hasta el nuevo flanco del reloj, pues las realimentaciones quedan cortadas en los biestables.



## 1.1 DEFINICIÓN DE UN SISTEMA SECUENCIAL (IV)

### VARIABLES

El número de **variables independientes** es el conjunto de:

- Entradas al sistema.
- Salidas de los biestables.
- El **estado interno** del circuito:
  - Es igual a la combinación de valores que tengan los biestables.
- A partir de las entradas al sistema y el estado interno se generarán:
  - Las salidas del circuito.
  - Las **propias entradas a los elementos de memoria (biestables)**.



## 1.1 DEFINICIÓN DE UN SISTEMA SECUENCIAL (V)

Variables: entrada  $x_i$ ,  $i=1\dots n$ ,

salida  $z_j$ ,  $j=1\dots m$

estado siguiente  $Y_k$ ,  $k=1\dots p$ ,

estado actual  $y_l$ ,  $l=1\dots q$ ,

Ecuaciones:

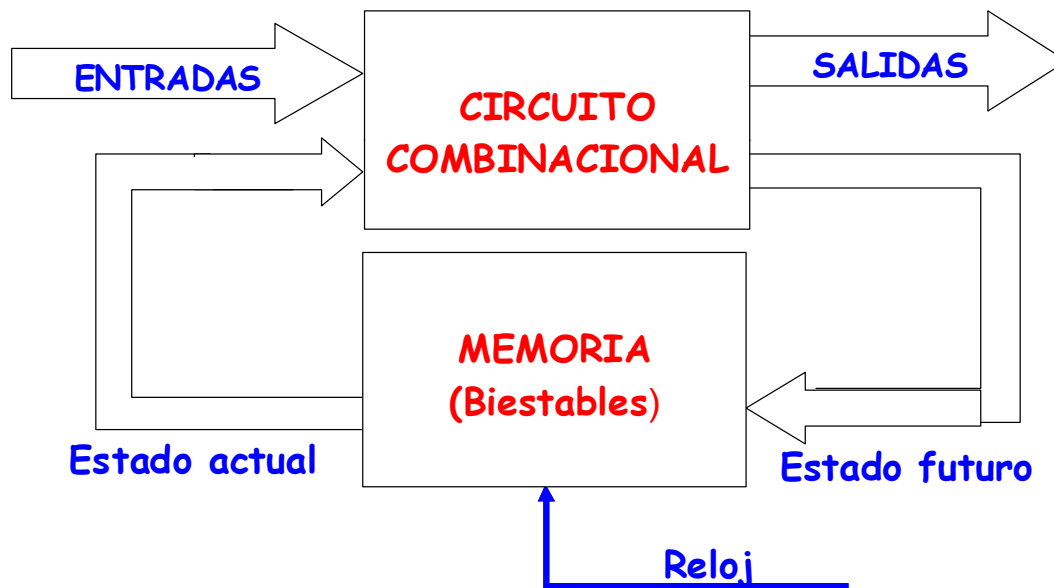
$$Z_j(t) = f_1 (x_1(t) x_2(t) \dots x_n(t), Y_1(t) Y_2(t) \dots Y_p(t))$$

$$Y_l(t) = f_2 (x_1(t) x_2(t) \dots x_n(t), y_1(t) y_2(t) \dots y_q(t))$$

$$y_k(t+1) = f_3 (Y_1(t) Y_2(t) \dots Y_q(t))$$



## 1.2 ESQUEMA GENERAL DE UN SISTEMA SECUENCIAL

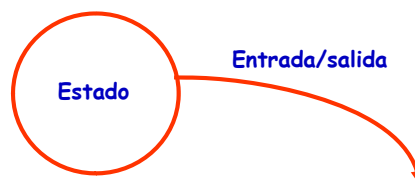


## 1.3 MÁQUINA DE ESTADOS FINITOS. AUTÓMATA

El modelo general de la máquina síncrona que estamos presentando es una "**Máquina de Estados Finitos**" o **FSM** (**Finite State Machine**), dado que el número de combinaciones de los valores de los biestables; es decir, en nuestro caso los estados internos del sistema, es finito.

## 1.3 MÁQUINA DE ESTADOS FINITOS. AUTÓMATA

La teoría de circuitos secuenciales se le denomina también teoría de autómatas, ya que estos ante una sucesión de valores de entrada que se presentan a lo largo del tiempo, proporcionan en cada instante una salida que depende de toda la secuencia de entradas llegadas hasta ese momento.



## 2. AUTÓMATAS DE MEALY Y DE MOORE (I)

### ❑ Autómata de MEALY

- ❑ Las salidas dependen de las entradas y de las variables de estado.

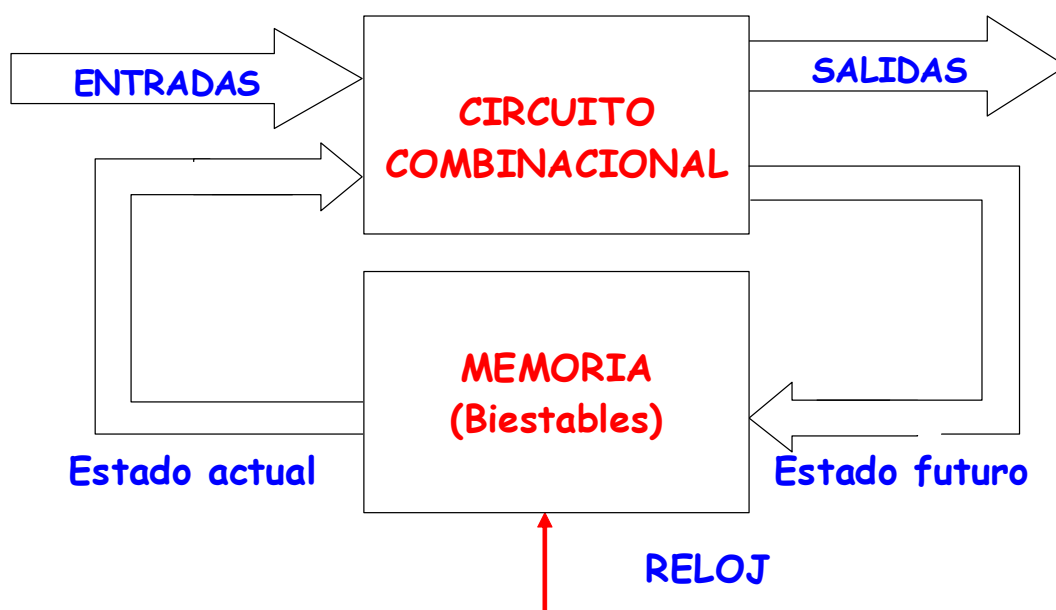
### ❑ Autómata de MOORE

- ❑ Las salidas no dependen directamente de las entradas, sino que éstas primeramente han de producir un cambio de estado; este cambio de estado determinará el nuevo valor de las salidas.
  - la salida está vinculada al estado, cambiando sincrónicamente con el reloj.
  - Las salidas no se modifican en las transiciones (arcos).
  - Cada cambio de la entrada, produce una modificación de las variables de excitación, que producirán con el reloj, un nuevo estado. Las salidas dependerán de este estado.

## 2. AUTÓMATAS DE MEALY Y DE MOORE (II)

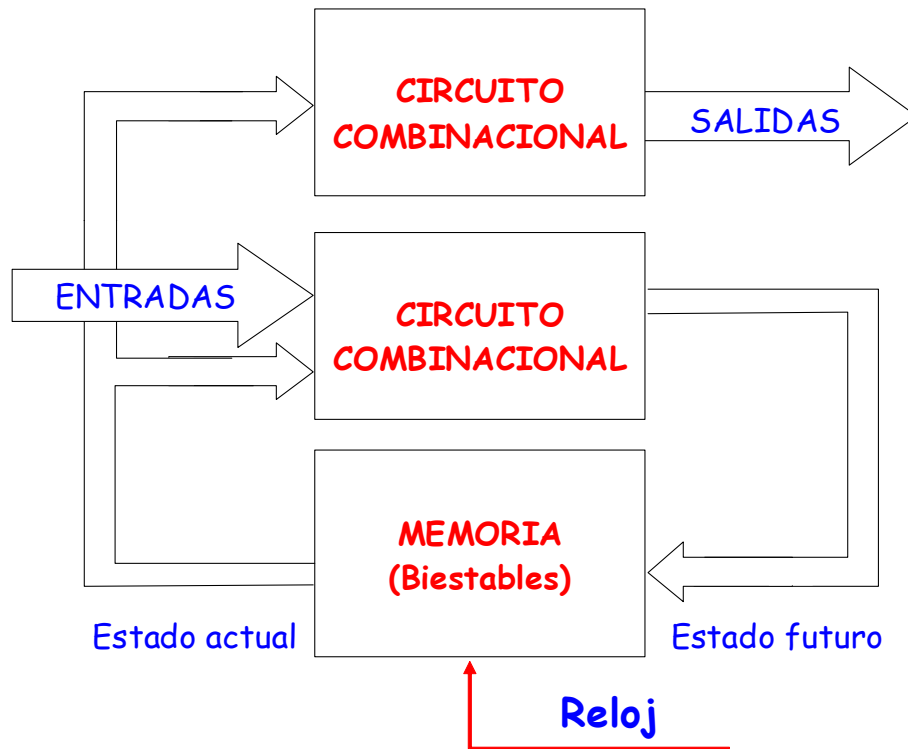
- ❑ Las máquinas de Moore, al cambiar las salidas de forma síncrona con el reloj, son en general, más estables que las máquinas de Mealy, ya que cualquier ruido transitorio que pudiera traer la entrada queda eliminado en las salidas.
- ❑ Las máquinas de Moore constituyen un caso particular de las máquinas de Mealy.

### 2.1 MODELO DE MEALY





## 2.2 MODELO DE MOORE



## 3. ESPECIFICACIÓN Y ETAPAS DE DISEÑO DE SISTEMAS SECUENCIALES (I)

El diseño de un sistema secuencial constará de dos etapas:

1. Determinación del número de estados necesarios, o lo que es equivalente, del número de biestables,  $N$ 
  - El número de estados  $P$  deberá cumplir:  
$$P \leq 2^N.$$
2. Construcción de la red combinacional, tal que para cada combinación del estado interno y las entradas proporcione las salidas requeridas y los valores que han de atacar a los biestables →
  - Estos últimos darán lugar, en el próximo ciclo de reloj, al nuevo estado.

## DISEÑO DE LA MEMORIA:

- Elección del tipo de autómata (Mealy o Moore)
- Determinación del número de estados internos, P.
- Codificación de los estados
- Descripción formal del problema (Tabla de estados, Diagrama de transiciones, tabla de transiciones, diagramas ASM,.....)
- Elección del tipo de biestable.
- Determinación del número de biestables, N.

$$P \leq 2^N$$



## DISEÑO DEL SISTEMA (RED) COMBINACIONAL

- Identificación de variables: entrada, salida, estado y excitación.
- Realización de la tabla de transiciones entre estados y la tabla de salida.
- Obtención de las funciones lógicas de salida y de entradas a los biestables.
- Realización física mediante elementos combinacionales (puertas lógicas, muxe's, .....)



### 3.1 TABLAS Y DIAGRAMAS DE TRANSICIÓN ENTRE ESTADOS (I)

La tabla de transición entre estados es directamente la especificación de la parte combinacional del sistema secuencial

En la tabla de transición entre estados, para cada estado y para combinación de las entradas se recoge cuál va a ser el próximo estado y qué valor van a tomar las señales de salida.

1. **Se nombran los estados.**
2. **Se codifican los estados:**
  1. Se cuenta el número de estados **distintos**.
  2. Se escoge el número adecuado de biestables.
  3. Se asigna un código (una combinación de variables) distinto a cada estado.



### 3.1 TABLAS Y DIAGRAMAS DE TRANSICIÓN ENTRE ESTADOS (II)

Un diagrama de transición de estados, diagramas de flujo, autómata es una representación que muestra las transiciones entre los estados, de acuerdo con las condiciones de conmutación. Se representa utilizando los siguientes símbolos:

#### □ **Círculos.**

- Indican un estado. Dentro del círculo aparecerá una etiqueta del tipo:

codificación del estado. (MEALY)

codificación del estado/salida del estado. (MOORE)

#### □ **Arcos.**

- Indican la transición de un estado origen a un estado destino por medio de una ocurrencia en alguna de sus entradas. El arco se acompaña de una etiqueta del tipo:

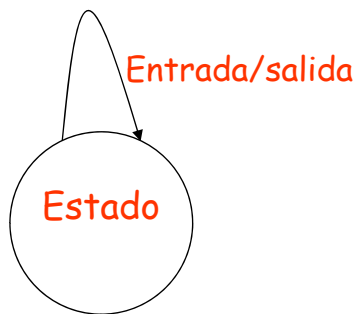
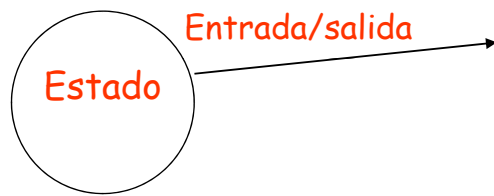
entrada o entradas. (MOORE)

entrada o entradas/salida o salidas. (MEALY)

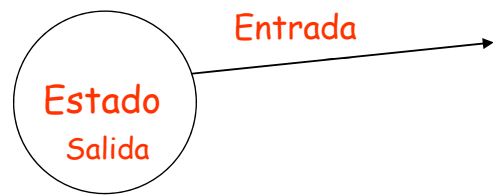


### 3.1 TABLAS Y DIAGRAMAS DE TRANSICIÓN ENTRE ESTADOS (III)

#### AUTÓMATA DE MEALY



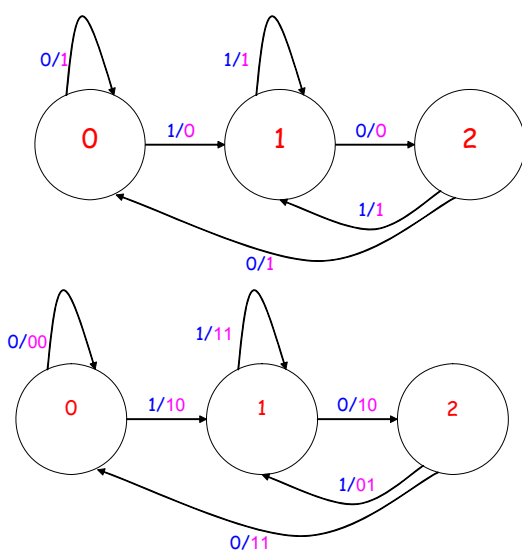
#### AUTÓMATA DE MOORE



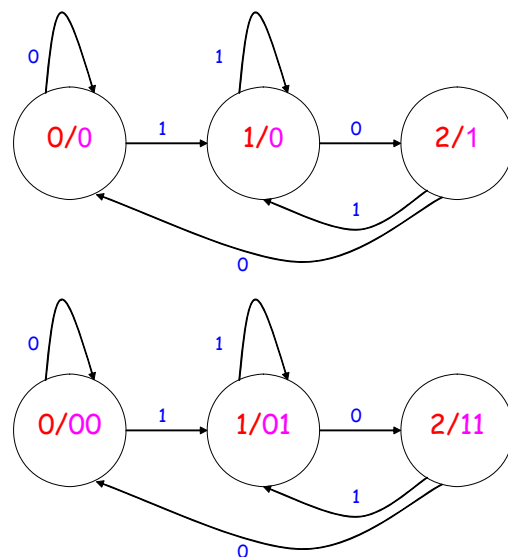
### 3.1 TABLAS Y DIAGRAMAS DE TRANSICIÓN ENTRE ESTADOS (IV)

#### EJEMPLOS:

##### Autómatas de MEALY



##### Autómatas de MOORE



## 3.2 EJEMPLOS DE CIRCUITOS SECUENCIALES (I)

1. Diseñese un detector de secuencia, que detecte la secuencia 101 y que una vez detectada permanezca en esa situación hasta que se inicialice. Impleméntese con biestables tipo D disparados por flanco de bajada, con señal CLEAR, y puertas lógicas necesarias para la red combinacional.

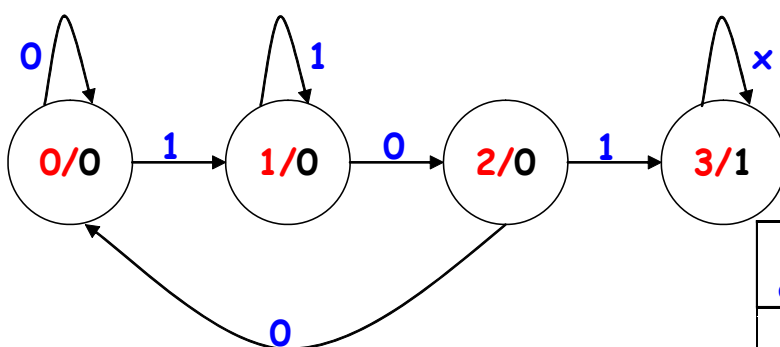
1. Realícese mediante un autómatas de Mealy.
2. Realícese mediante un autómatas de Moore.
3. Implementación canónica del detector
4. Descripción mediante diagramas ASM
5. Modelado VHDL en los tres estilos de descripción



25

### 3.2.1 DETECTOR DE SECUENCIA "101" (II)

#### Autómata de MOORE



#### Tabla de transición entre estados

Estado actual (t)		Ent (t)	Estado sig.(t+1)		Sal (t)
Q <sub>1</sub>	Q <sub>0</sub>	E	D <sub>1</sub>	D <sub>0</sub>	S
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	1	1	1

#### Diagrama de transición entre estados (flujo)

Entradas: E (una entrada)  
 Salidas: S (una salida)  
 Estados: 4 estados  $4 = 2^p \rightarrow p=2$   
 N° Biestables: 2



26

### 3.2<sub>1</sub> DETECTOR DE SECUENCIA "101" (III)

D <sub>1</sub>	E	
Q <sub>1</sub> Q <sub>0</sub>	0	1
00	0	0
01	1	0
11	1	1
10	0	1

D <sub>0</sub>	E	
Q <sub>1</sub> Q <sub>0</sub>	0	1
00	0	1
01	0	1
11	1	1
10	0	1

$$D_0 = E + Q_1Q_0$$

$$S = Q_1Q_0$$

$$D_1 = \bar{E}Q_0 + EQ_1$$



27

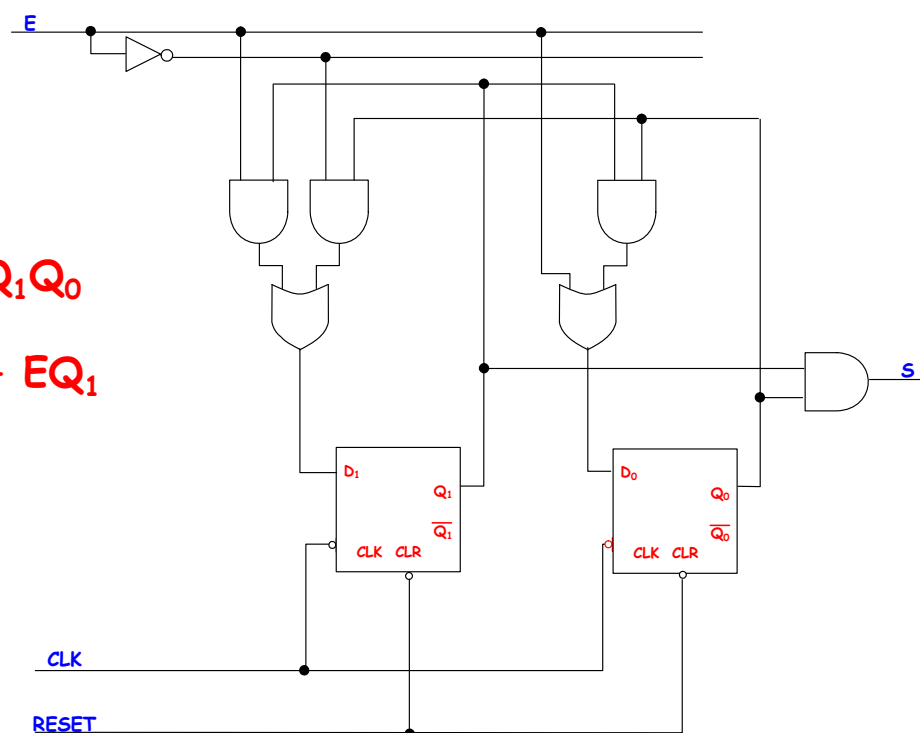
### 3.2<sub>1</sub> DETECTOR DE SECUENCIA "101" (IV)

Implementación con biestables tipo D y puertas lógicas

$$D_0 = E + Q_1Q_0$$

$$D_1 = \bar{E}Q_0 + EQ_1$$

$$S = Q_1Q_0$$



28

### 3.2<sub>1</sub> DETECTOR DE SECUENCIA "101" (V)

#### Autómata de MEALY

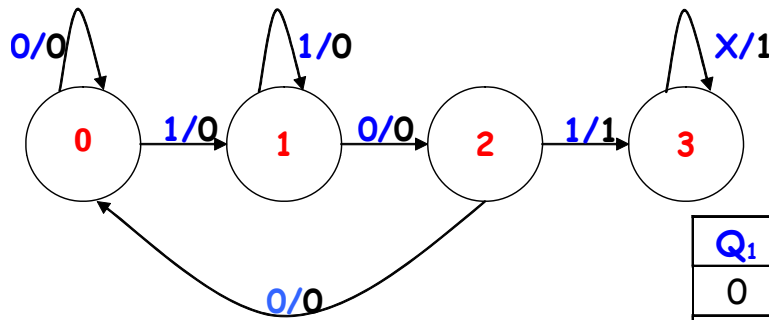


Tabla de transición entre estados

$Q_1$	$Q_0$	$E$	$D_1$	$D_0$	$S$
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Diagrama de transición entre estados (flujo)

Entradas:  $E$  (una entrada)  
 Salidas:  $S$  (una salida)  
 Estados: 4 estados  $4 = 2^p \rightarrow p=2$   
 N° Biestables: 2



### 3.2<sub>1</sub> DETECTOR DE SECUENCIA "101" (VI)

$$D_0 = E + Q_1 Q_0$$

$$D_1 = \bar{E} Q_0 + E Q_1$$

$$S = Q_1 E + Q_1 Q_0$$

$S$	$E$	
$Q_1 Q_0$	0	1
00	0	0
01	0	0
11	1	1
10	0	1



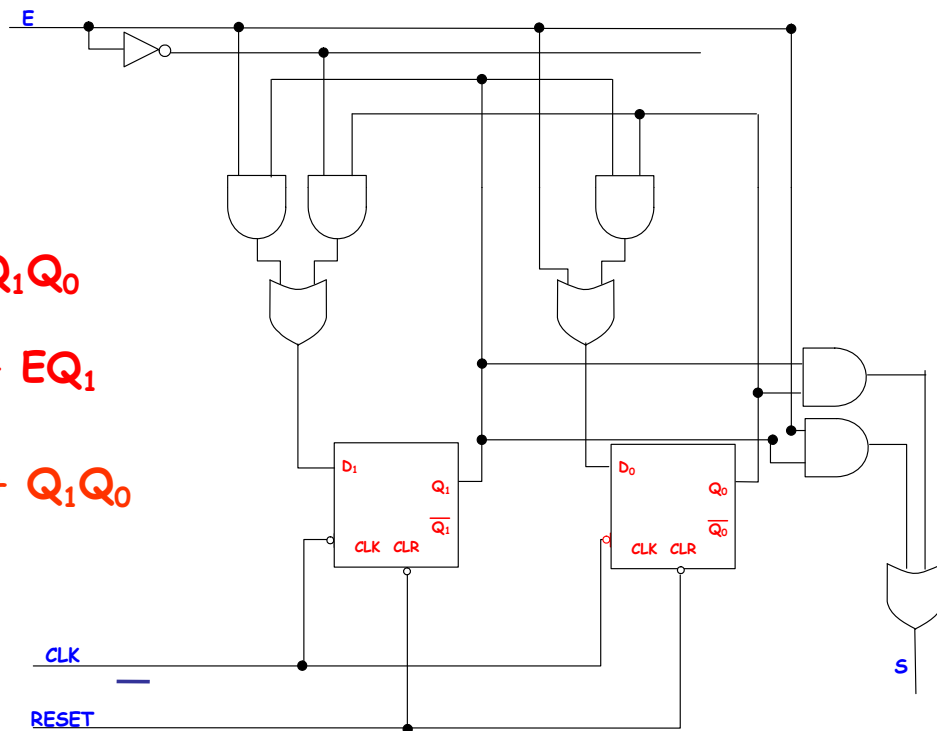
### 3.2<sub>1</sub> DETECTOR DE SECUENCIA "101" (VII)

Implementación con biestables tipo D y puertas lógicas

$$D_0 = E + Q_1 Q_0$$

$$D_1 = \bar{E} Q_0 + E Q_1$$

$$S = Q_1 E + Q_1 Q_0$$



31

### 3.2<sub>2</sub> EJEMPLOS DE CIRCUITOS SECUENCIALES (II)

2. Diseñese un contador hasta 10 (módulo 10) síncrono, mediante un autómata de Moore, e impleméntese con biestables tipo D disparados por flanco de bajada, con señal CLEAR, y puertas lógicas para la red combinacional.

Autómata de Moore:

Entradas: no tiene entradas

Estados: hasta 10 se codifica con  
4 bits → 4 biestables

10 estados;  $10 \leq 2^p \Rightarrow p=4$

Nº Biestables: 4

Salidas: es el estado

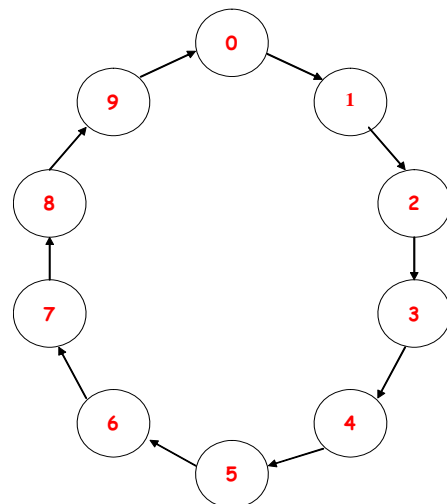


Diagrama de flujo

32



### 3.2<sub>2</sub> Contador módulo 10 (II)

ESTADO ACTUAL				ESTADO PRÓXIMO			
Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x



33

### 3.2<sub>2</sub> Contador módulo 10 (III)

D <sub>0</sub>	Q <sub>1</sub> Q <sub>0</sub>			
Q <sub>3</sub> Q <sub>2</sub>	00	01	11	10
00	1	0	0	1
01	1	0	0	0
11	x	x	x	x
10	1	0	x	x

D <sub>1</sub>	Q <sub>1</sub> Q <sub>0</sub>			
Q <sub>3</sub> Q <sub>2</sub>	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	x	x	x	x
10	0	0	x	x

$$D_0 = \overline{Q_0}$$

$$D_1 = Q_1 \overline{Q_0} + \overline{Q_3} \overline{Q_1} Q_0$$

$$D_2 = Q_2 \overline{Q_1} + Q_2 \overline{Q_0} + \overline{Q_2} Q_1 Q_0$$

$$D_3 = Q_3 \overline{Q_0} + Q_2 Q_1 Q_0$$

D <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>			
Q <sub>3</sub> Q <sub>2</sub>	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	x	x	x	x
10	0	0	x	x

D <sub>3</sub>	Q <sub>1</sub> Q <sub>0</sub>			
Q <sub>3</sub> Q <sub>2</sub>	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	x	x	x	x
10	1	0	x	x

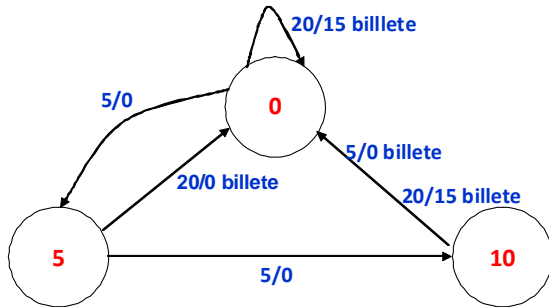
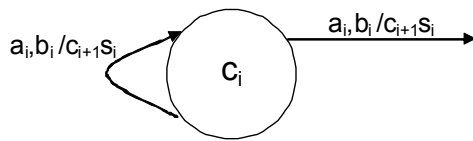


34

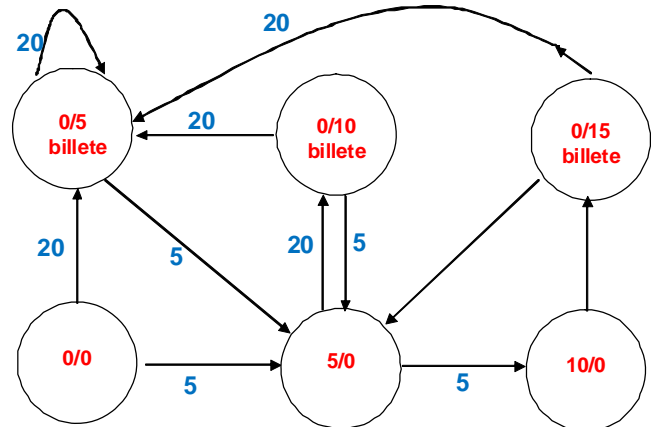
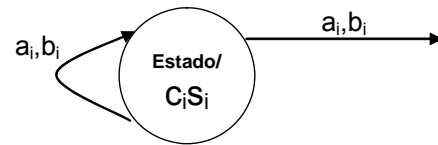


### 3.2<sub>3</sub> MÁQUINA EXPENDEDORA (I)

#### EJ 3 Autómata de Mealy



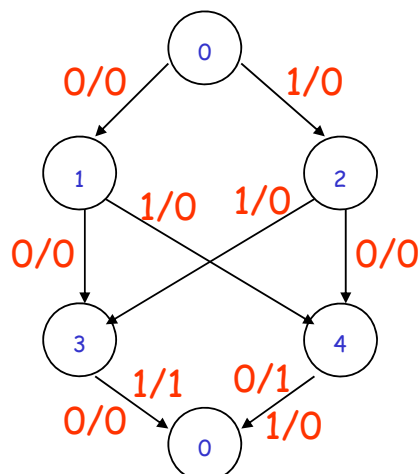
#### Autómata de Moore



37

### 3.2<sub>4</sub> RED DE DETECCIÓN DE NÚMERO DE 1'S (I)

**EJ4: Variables:** entrada  $w$   
 salida  $z$   
 estados: 5 estados;  $5 \leq 2^p \rightarrow p=3$   
 estado  $Q_A Q_B Q_C$   
 excitación  $D_A D_B D_C$



3 biestables tipo D ↓



38

### 3.2<sub>4</sub> RED DE DETECCIÓN DE NÚMERO DE 1'S (II)

Estado actual (t)			Entrada (t)	Estado próximo $Q_i(t+1)=D_i$			Salida (t)
$Q_A$	$Q_B$	$Q_C$	$W$	$D_A$	$D_B$	$D_C$	$Z$
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	1	1	0
0	0	1	1	1	0	0	0
0	1	0	0	1	0	0	0
0	1	0	1	0	1	1	0
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				



39

## 4. IMPLEMENTACIÓN CANÓNICA DE SS: CONSTRUCCIÓN CON BIESTABLES, MUX'ES Y DECODIFICADORES (I)

Utilización de circuitos que presenten regularidad en su estructura: multiplexores y decodificadores.

Consideraciones para la implementación del sistema combinacional:

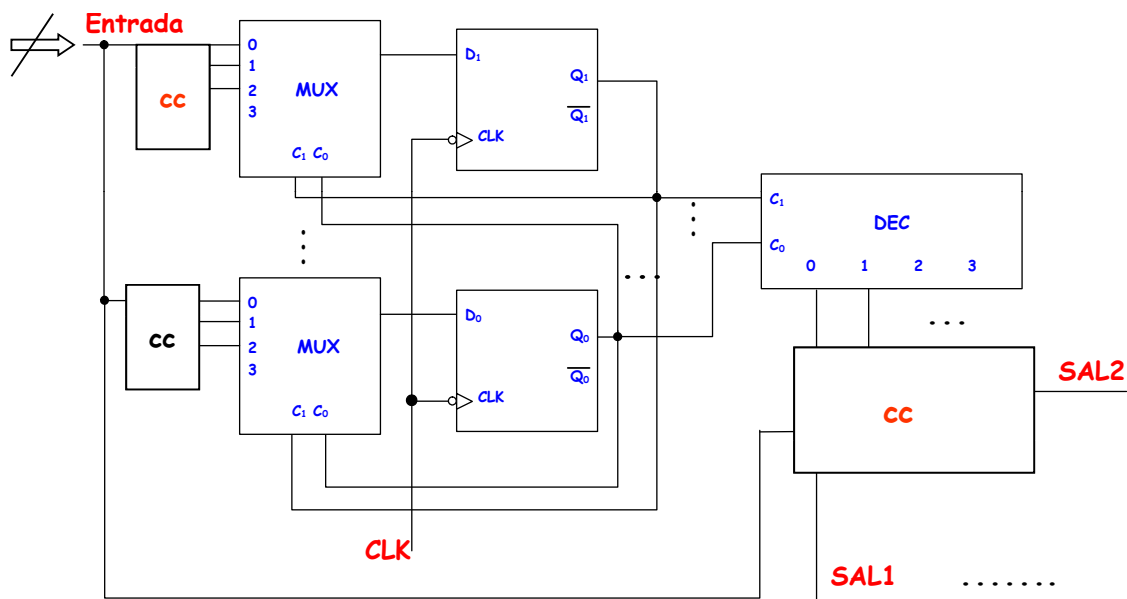
- ❑ **Decodificación de estados:** las variables de estado se toman como entradas a un decodificador → las salidas serán los estados
- ❑ **Realización del secuenciamiento de los estados mediante multiplexores** → el valor de las variables que determinan el estado siguiente, variables de excitación, solamente depende de las entradas al sistema. Esto supone la utilización de un mux, por cada biestable →
  - Las entradas de control de los mux'es: variables de estado
  - Las entradas de los mux'es: funciones combinacionales de las entradas del sistema



40

## 4. IMPLEMENTACIÓN CANÓNICA DE SS: CONSTRUCCIÓN CON BIESTABLES, MUX'ES Y DECODIFICADORES (II)

DIAGRAMA DE BLOQUES



### 4.1 EJEMPLOS DE IMPLEMENTACIÓN (I)

1. Diseñese un detector de secuencia, que detecte la secuencia 101 y que una vez detectada permanezca en esa situación hasta que se inicialice. Impleméntese con biestables tipo D disparados por flanco de bajada, con señal CLEAR, y puertas lógicas necesarias para la red combinacional.
1. Realícese mediante un autómata de Mealy.
2. Realícese mediante un autómata de Moore.
3. Implementación canónica del detector.
4. Descripción mediante diagramas ASM
5. Modelado VHDL en los tres estilos de descripción



## 5. ESPECIFICACIÓN de SS MEDIANTE DIAGRAMAS ASM (II)

- ❑ Estos diagramas expresan detalladamente el algoritmo que se debe de seguir para resolver un determinado problema.
- ❑ Indican:
  - ❶ Los estados por los que puede atravesar el sistema.
  - ❷ Las condiciones que producen transiciones entre estados. Entre dos estados hay un ciclo de reloj.
  - ❸ Las señales de salida que se deben producir en cada estado.



## 5. ESPECIFICACIÓN de SS MEDIANTE DIAGRAMAS ASM (III)

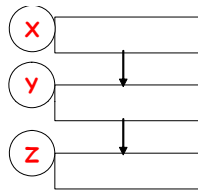
Un diagrama ASM consta de los siguientes símbolos:

- ❑ Rectángulos (disponen de un pequeño círculo en el ángulo superior izquierdo, para incluir el nombre).
  - Identifican los estados.
  - Dentro del rectángulo se especifican aquellas salidas que se activan siempre que nos encontremos en dicho estado, con independencia de los valores que tomen las entradas.
- ❑ Rombos o hexágonos.
  - Permiten expresar bifurcaciones en el flujo de control.
- ❑ Óvalos.
  - Indican las salidas condicionales.
    - Siempre se presentan a continuación de una bifurcación.

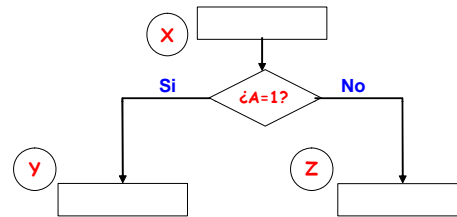


## 5. ESPECIFICACIÓN de SS MEDIANTE DIAGRAMAS ASM (IV)

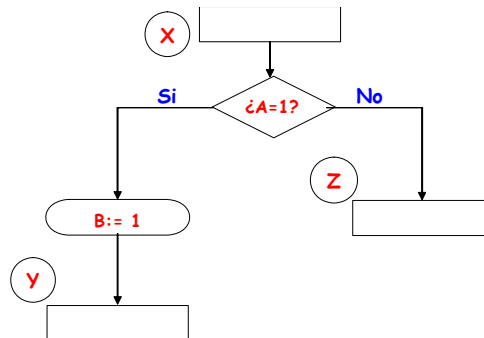
1. Secuenciamiento de estados.



2. Bifurcaciones en el flujo de control del algoritmo.

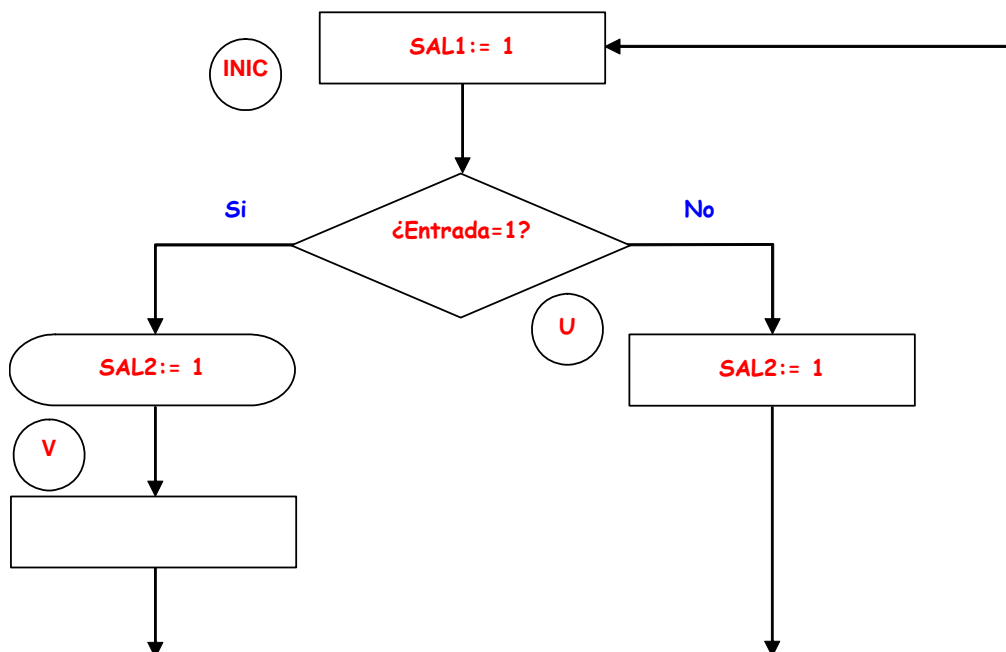


3. Salidas condicionales.



## 5. ESPECIFICACIÓN de SS MEDIANTE DIAGRAMAS ASM (V)

### Ejemplo





## 5. ESPECIFICACIÓN de SS MEDIANTE DIAGRAMAS ASM (VI)

### COMENTARIOS:

1. Dado que hay salidas condicionales el autómata es de Mealy.
2. Tenemos 3 estados → necesitamos 2 biestables.
3. Variables:  
 entrada **E**  
 salidas **Sal<sub>1</sub> Sal<sub>2</sub>**  
 estados **inic, u, v**
4. Utilizaremos implementación canónica y biestables tipo D disparados por flanco de bajada.



## 5. ESPECIFICACIÓN de SS MEDIANTE DIAGRAMAS ASM (VII)

Tabla de transición entre estados y de salida

Estado actual			Entradas	Estado siguiente			Salidas	
Nombre	Q <sub>1</sub>	Q <sub>0</sub>	Entrada		D <sub>1</sub>	D <sub>0</sub>	Sal1	Sal2
INIC:	0	0	0	U:	0	1	1	0
INIC:	0	0	1	V:	1	0	1	1
U:	0	1	x	INIC:	0	0	0	1
V:	1	0	x	INIC:	0	0	0	0

$$D_0 = \text{INIC} \cdot E = \overline{Q_1} \overline{Q_0} \overline{E}$$

$$D_1 = \text{INIC} \cdot E = \overline{Q_1} \overline{Q_0} E$$

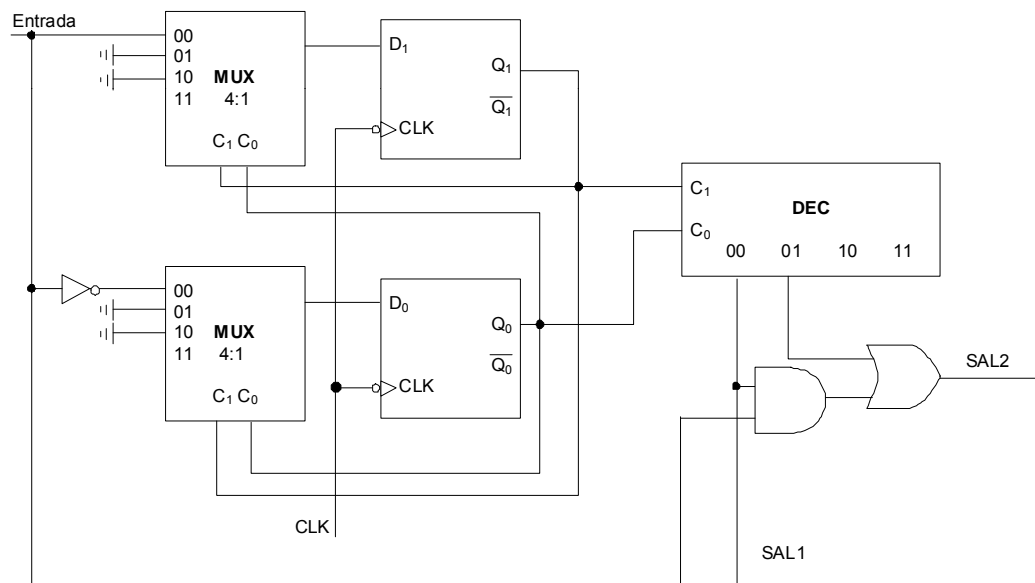
$$\text{Sal1} = \text{INIC} = \overline{Q_1} \overline{Q_0}$$

$$\text{Sal2} = \text{INIC} \cdot E + U = \overline{Q_1} \overline{Q_0} E + \overline{Q_1} Q_0$$



## 5. ESPECIFICACIÓN de SS MEDIANTE DIAGRAMAS ASM (VIII)

### Implementación canónica del ejemplo



51

### 5.1<sub>1</sub> DETECTOR DE SECUENCIA "101" (IX)

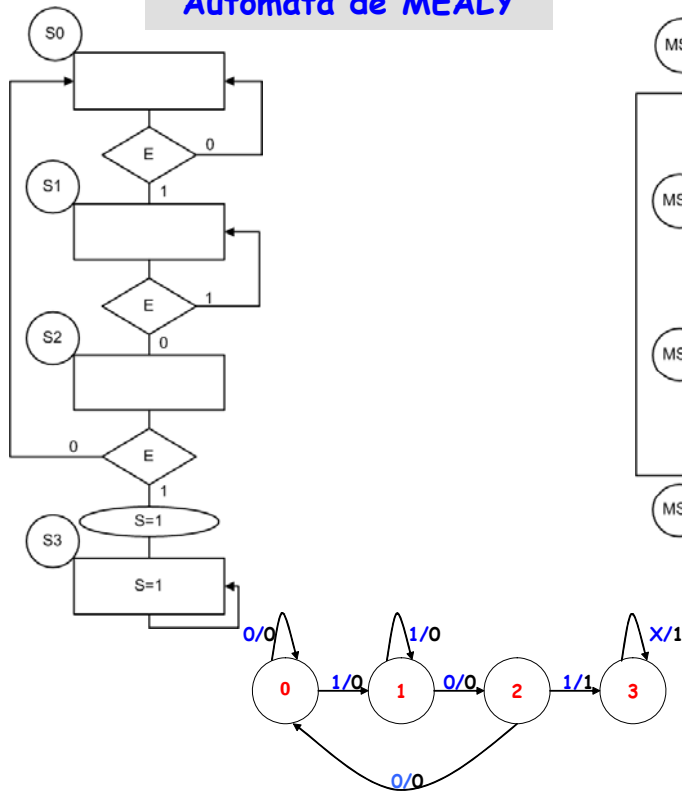
1. Diseñese un detector de secuencia, que detecte la secuencia 101 y que una vez detectada permanezca en esa situación hasta que se inicialice. Impleméntese con biestables tipo D disparados por flanco de bajada, con señal CLEAR, y puertas lógicas necesarias para la red combinacional.
1. Realícese mediante un autómata de Mealy.
2. Realícese mediante un autómata de Moore.
3. Implementación canónica del detector.
4. Descripción mediante diagramas ASM
5. Modelado VHDL en los tres estilos de descripción



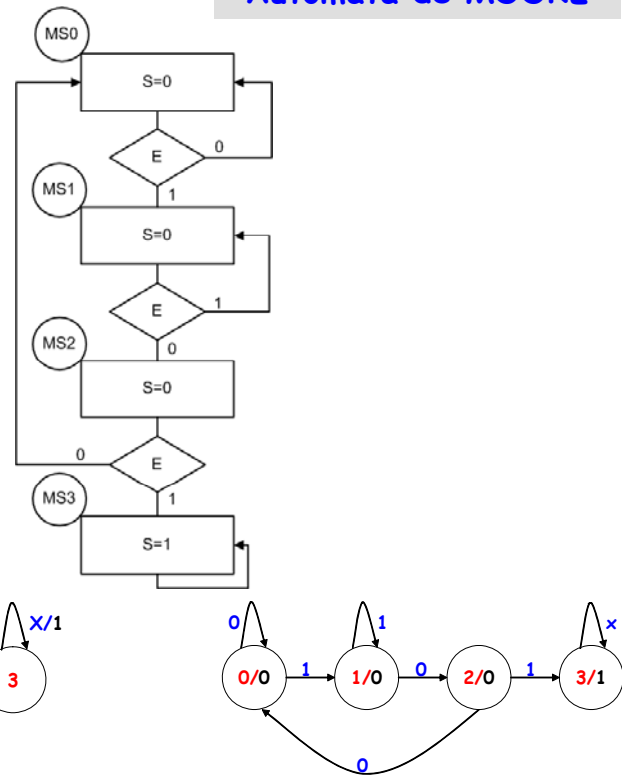
52

## 5.1<sub>1</sub> DETECTOR DE SECUENCIA "101" (X)

Autómata de MEALY



Autómata de MOORE



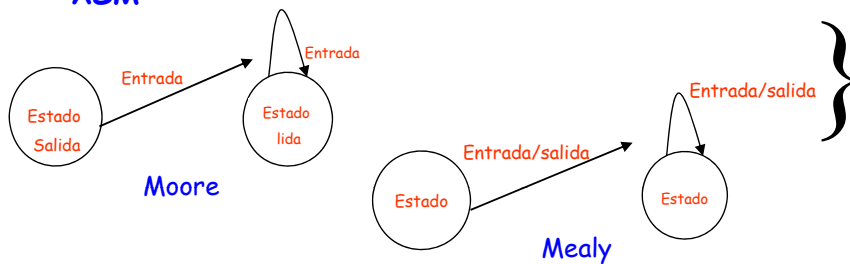
## 6. MODELADO VHDL DE MÁQUINAS DE ESTADOS FINITOS (FSM) (I)

- Modelos de Moore y Mealy
- Metodología de diseño de FSM
  - Modelo Comportamiento (Algorítmico)
    - Señales de Reset
    - Varios PROCESS
  - Modelo Flujo de datos (Transferencia entre registros)
  - Modelo estructural

## 4. METODOLOGÍA DE DISEÑO DE (FSM'S) (II)

Niveles de Descripción  
asociados con los HDLs

### 1) Diagrama o Tabla de estados, diagrama ASM



Comportamiento o  
algorítmico

### 2) Ecuaciones del estado futuro y salidas

$$D1(t+1) = C1(t) A(t) + C0(t) A(t)' + \dots$$

$$D0(t+1) = C1(t) B(t) + C0(t) B(t)' + \dots$$

$$X(t) = \dots \quad Y(t) = \dots$$

Flujo de datos o  
Transferencia entre  
registros

### 3) Implementación del circuito: instanciación de componentes y conexonado

Dispositivos lógicos (puertas: NOT, AND... MUX, DECODERS, ....)

Elementos de memoria (biestables)

Estructural



55

## 4. MODELO ALGORÍTMICO DE UNA FSM (III)

### • Consideraciones Generales:

- El estilo del código tiene que ser independiente de la máquina de estados que se diseñe.
- Diferentes formas de modelar la misma máquina:
  - Número de procesos PROCESS
  - Sentencias IF o CASE
  - Uso de señales o de variables
  - El uso de la sentencia WAIT o lista de sensibilidad
- ¿Debe ser inicializada la máquina en un estado válido? ¿Hay que utilizar señal de reset?
- Portabilidad del código



56

## 4. MODELO ALGORÍTMICO DE UNA FSM (IV)

### • MODELO DE MEALY

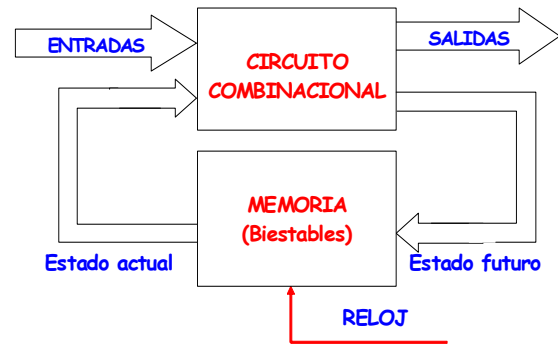
- DOS PROCESOS: uno para la generación del estado futuro y de la salida y otro para la actualización de la memoria

Salida, futuro: PROCESS (entrada, e\_actual)

```
salida <=  
e_futuro <=
```

Actual: PROCESS (reset, klok)

```
e_actual <=
```



## 4. MODELO ALGORÍTMICO DE UNA FSM (V)

### • MODELO DE MOORE

- TRES PROCESOS: uno para la generación del estado futuro otro para la generación de la salida y el tercero para la actualización de la memoria

Salida: PROCESS (e\_actual)

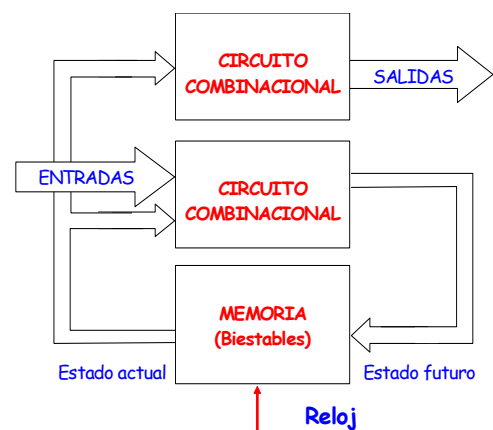
```
salida <=
```

Futuro: PROCESS (entrada, e\_actual)

```
e_futuro <=
```

Actual: PROCESS (reset, klok)

```
e_actual <=
```



## 4. MODELADO ALGORÍTMICO DE UNA FSM (VI)

- Uso de señales o de variables: los estados deben definirse con un tipo de **datos enumerado** y deben ser **señales** para que puedan comunicar procesos
- Sentencias IF o CASE: los estados deben modelarse con sentencia **CASE** y las transiciones con sentencias **IF**
- ¿Debe ser inicializada la máquina en un estado válido?: es recomendable que se inicialice en un **estado inicial**
- ¿Hay que utilizar señal de reset?: es recomendable
- La señales que activan PROCESS deben ir en la **lista de sensibilidad** o en **WAIT**:
  - ❑ En la generación de las salidas y/o el estado futuro, deben incluirse **todas las señales** que entran en el bloque
  - ❑ En el bloque de memoria se incluirá siempre **el reloj** y la señal **reset**, si es asíncrona



## 4. MODELADO EN FLUJO DE DATOS DE UNA FSM (VII)

- Se parte de las ecuaciones lógicas
- Se utilizan sentencias de **asignación concurrente de señal**
- ¿Hay que utilizar señal de reset?: es recomendable
- La señales que activan PROCESS deben ir en la **lista de sensibilidad** o en **WAIT**:
  - Generación de las salidas y/o del estado futuro (entradas a los biestables)
    - $D_i \leftarrow$
    - $S_i \leftarrow$
  - Generación del estado actual (**reloj y reset**)
    - $Q_i \leftarrow$  ( ...reset, rising\_edge, fallinf\_edge.....)



## 4. MODELADO ESTRUCTURAL DE UNA FSM (VIII)

- Se debe disponer, en la librería **WORK**, de los componentes necesarios para la implementación.
- Se instancian y se conectan con las señales adecuadas
  - Biestables, puertas, muxe's, decoder's ,.....
  - **POR MAP**



61

## 6.1 EJEMPLOS DE MODELADO VHDL (XI)

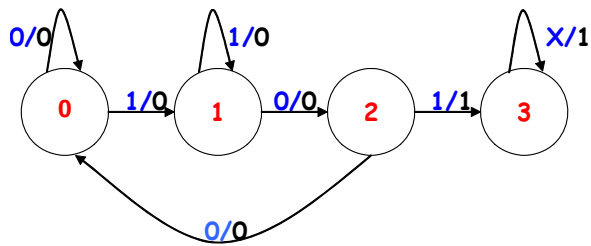
1. Diseñese un detector de secuencia, que detecte la secuencia 101 y que una vez detectada permanezca en esa situación hasta que se inicialice. Impleméntese con biestables tipo D disparados por flanco de bajada, con señal CLEAR, y puertas lógicas necesarias para la red combinacional.
  1. Realícese mediante un autómata de Mealy.
  2. Realícese mediante un autómata de Moore.
  3. Implementación canónica del detector
  4. Descripción mediante diagramas ASM
  5. **Modelado VHDL en los tres estilos de descripción**



62

## 6.1<sub>1</sub> DETECTOR DE SECUENCIA "101" (XII)

### AUTÓMATA DE MEALY



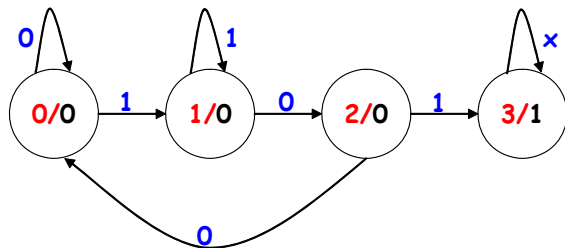
$$D_1 = \overline{Q_1}Q_0\overline{E} + Q_1\overline{Q_0}E + Q_1Q_0(E + \overline{E}) = Q_0\overline{E} + Q_1E$$

$$D_0 = \overline{Q_1}\overline{Q_0}E + \overline{Q_1}Q_0E + Q_1\overline{Q_0}E + Q_1Q_0(E + \overline{E}) = E + Q_1Q_0$$

$$S_{Mealy} = Q_1\overline{Q_0}E + Q_1Q_0(\overline{E} + E) = Q_1E + Q_1Q_0$$

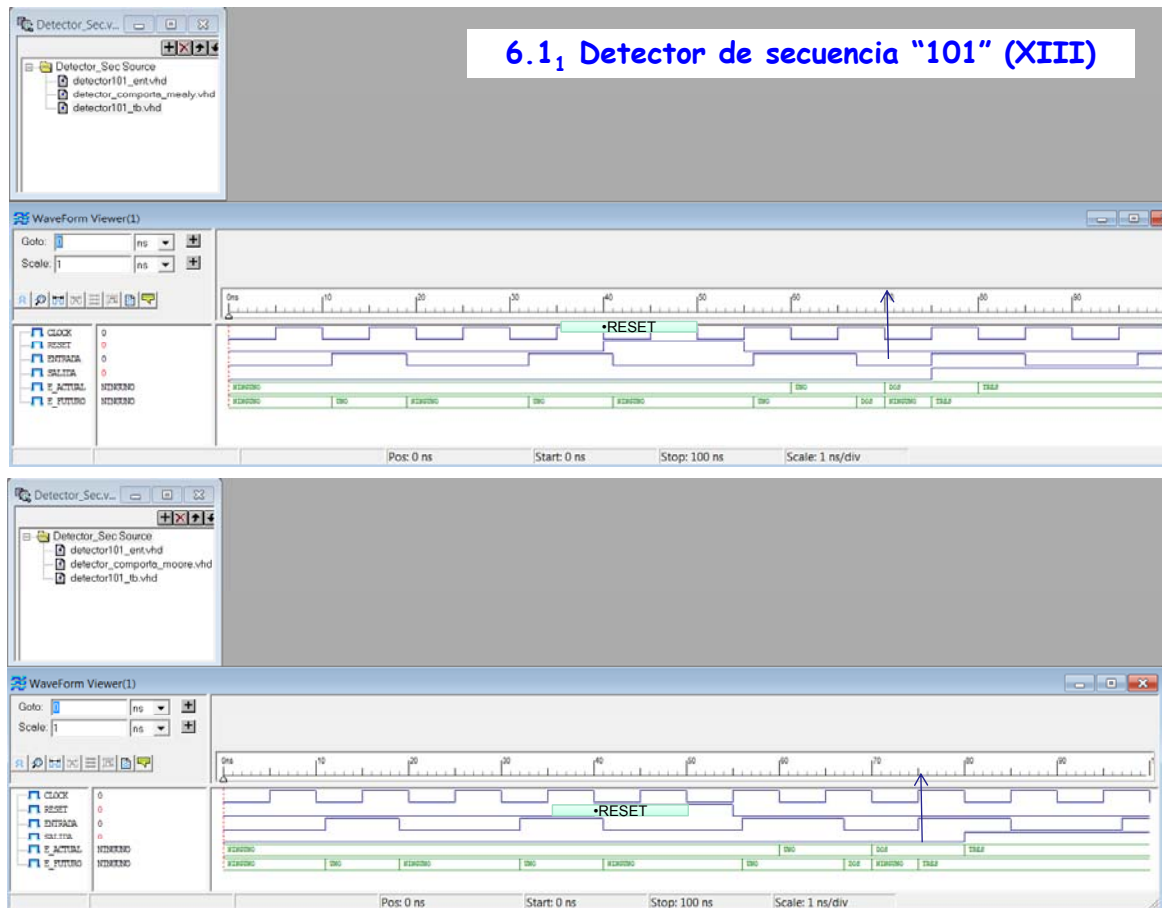
$$S_{Moore} = Q_1Q_0(\overline{E} + E) = Q_1Q_0$$

### AUTÓMATA DE MOORE



63

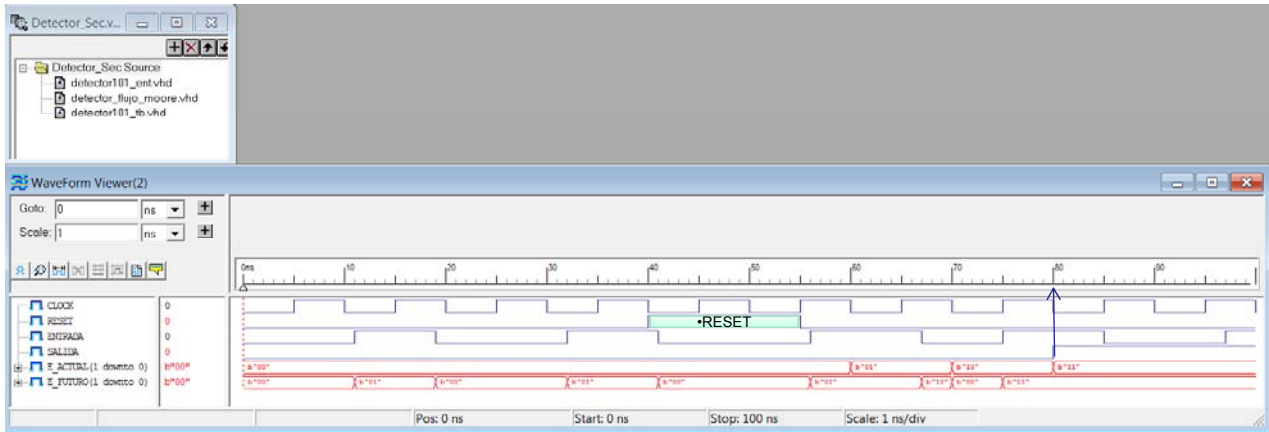
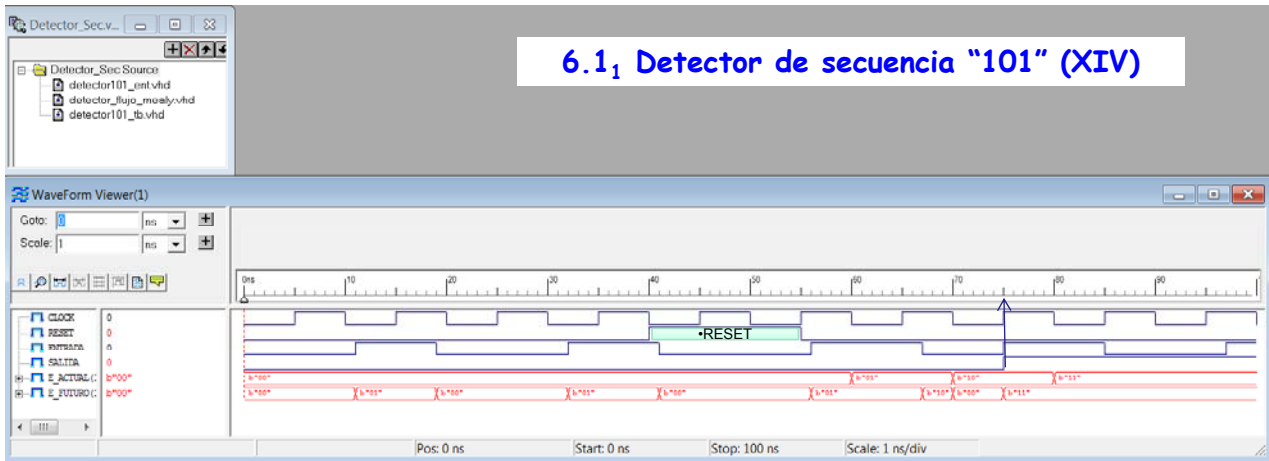
## 6.1<sub>1</sub> Detector de secuencia "101" (XIII)



64



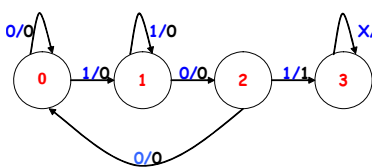
## 6.1<sub>1</sub> Detector de secuencia "101" (XIV)



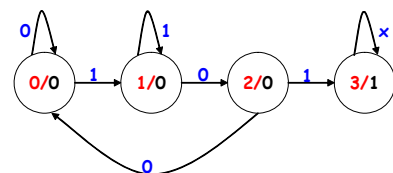
65

## 6.1<sub>1</sub> Detector de secuencia "101" (XV)

### AUTÓMATA DE MEALY



### AUTÓMATA DE MOORE

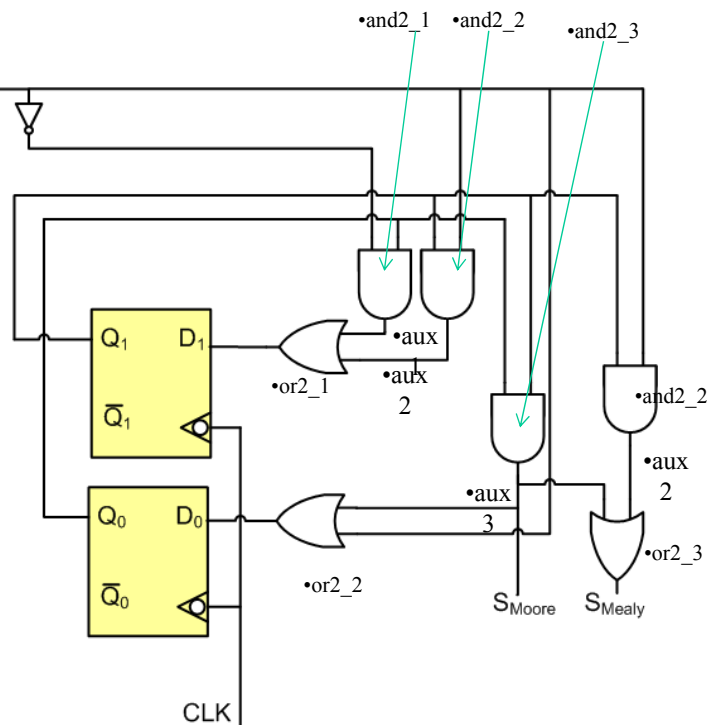


$$D_1 = \overline{Q_1}Q_0\overline{E} + Q_1\overline{Q_0}E + Q_1Q_0(E + \overline{E}) = Q_0\overline{E} + Q_1E$$

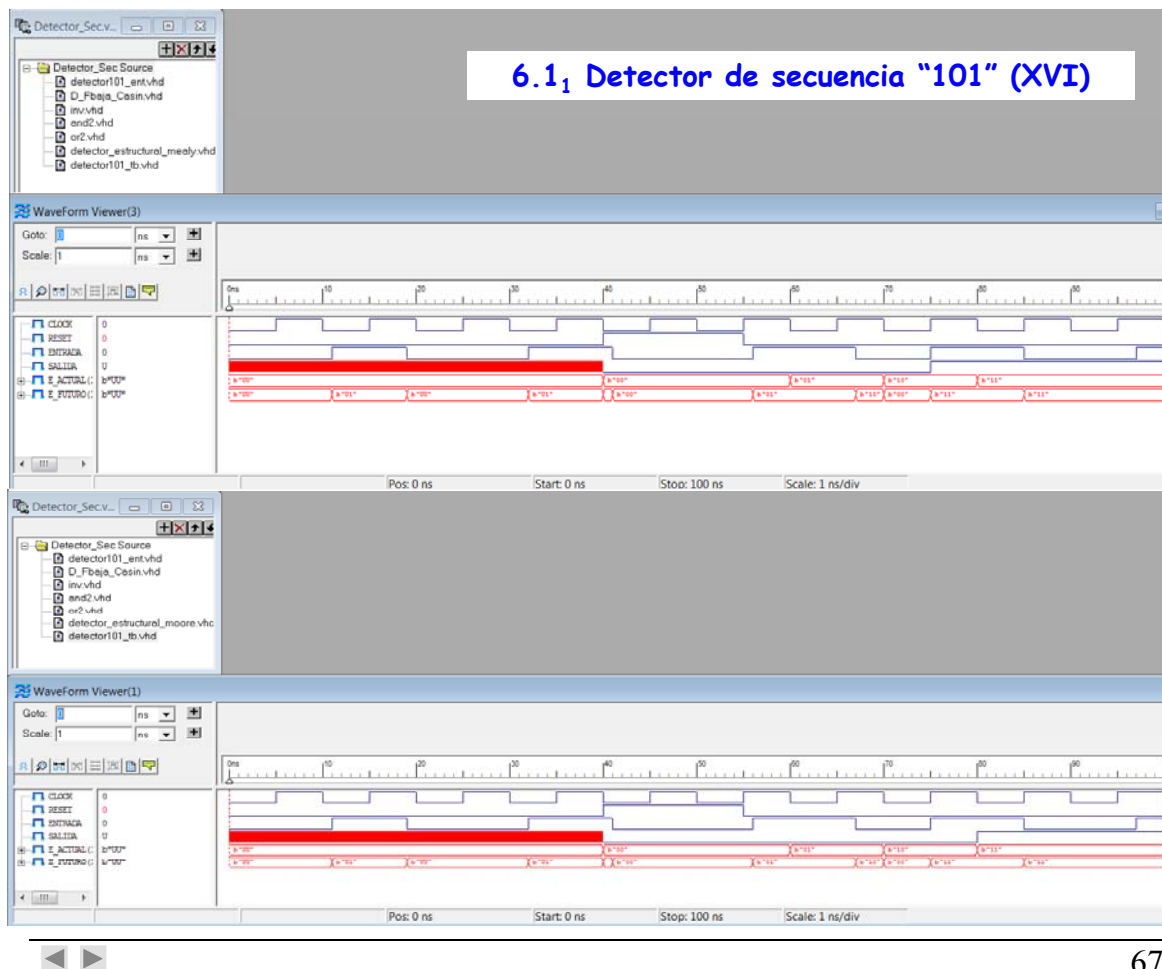
$$D_0 = \overline{Q_1}\overline{Q_0}E + \overline{Q_1}Q_0\overline{E} + Q_1\overline{Q_0}E + Q_1Q_0(E + \overline{E}) = E + Q$$

$$S_{Mealy} = Q_1\overline{Q_0}E + Q_1Q_0(\overline{E} + E) = Q_1E + Q_1Q_0$$

$$S_{Moore} = Q_1Q_0(\overline{E} + E) = Q_1Q_0$$



66



67

## 7. ANÁLISIS Y SÍNTESIS DE SISTEMAS SECUENCIALES (I)

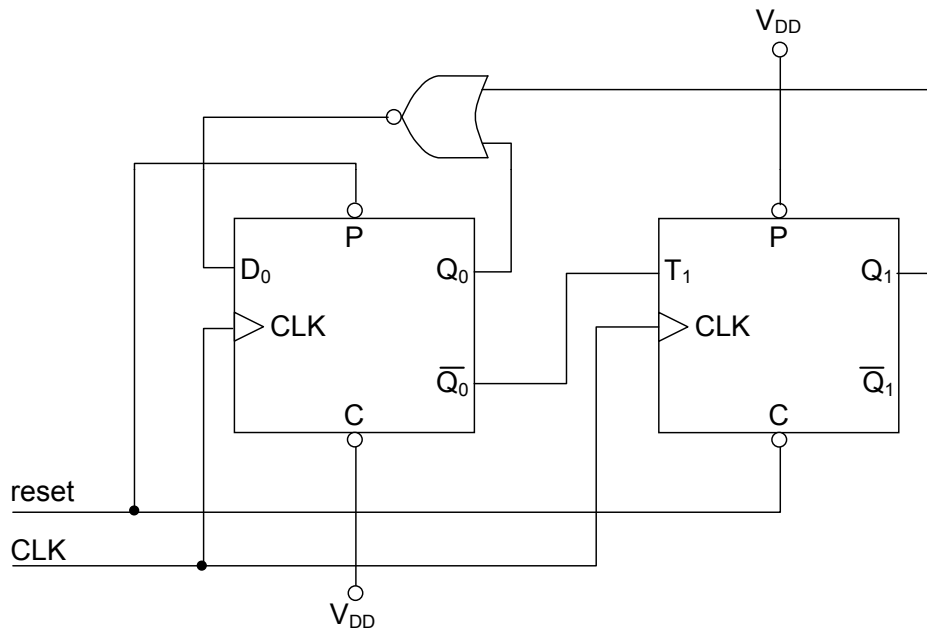
Dos formas contrapuestas de abordar el estudio de los sistemas secuenciales:

- Síntesis o construcción de sistemas secuenciales: partiendo de una especificación encontrar el sistema cuya implementación cumpla los requerimientos exigidos.
- Análisis de sistemas secuenciales: a la vista de una implementación, indicar la operación que realiza

68

## 7. ANÁLISIS Y SÍNTESIS DE SISTEMAS SECUENCIALES (II)

### EJEMPLOS de ANÁLISIS



## 7. ANÁLISIS Y SÍNTESIS DE SISTEMAS SECUENCIALES (III)

### EJEMPLOS de ANÁLISIS (II)

